



International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: <https://www.svedbergopen.com/>



Research Paper

Open Access

An Explainable Semantic Web and Machine Learning Framework For Intrusion Prediction in Internet of Vehicles

Naveen Joshi¹, Dr. Nirmal Kaur²

¹ Research Scholar, Sant Baba Bhag Singh University, Punjab, India.

² Associate Professor, Sant Baba Bhag Singh University, Punjab, India.

Abstract—The Internet of Vehicles (IoV) combines in-vehicle networks, roadside infrastructure, edge/cloud services and heterogeneous wireless links, creating a broad and dynamic attack surface. Machine-learning intrusion detection can classify abnormal traffic, but a class label alone may not explain which vehicle asset, context or security evidence supports the decision. This paper presents an explainable framework that combines Semantic Web knowledge representation, explicit rule traces and Random Forest classification for IoV intrusion assessment. The ontology models vehicles, electronic control units (ECUs), roadside units, communication events, observations, context, vulnerabilities, attack patterns and intrusion assessments, while RDF/OWL relationships preserve the evidence path behind an alert. The reported prototype contains a source-reported 1,200-record, seven-feature binary dataset and a 75/25 stratified split. On the 300-record test set, Random Forest obtained 91.33% accuracy, 92.59% intrusion precision, 84.75% recall and 88.50% F1-score (TN=174, FP=8, FN=18, TP=100). Packet rate (0.3252), source reputation (0.2694) and failed authentication count (0.1556) were the three most influential features. A worked semantic rule trace demonstrates how high packet rate, repeated authentication failure, abnormal beacon deviation and low reputation yield an interpretable risk assessment. The current evidence supports explainable current-event classification; advance attack-onset prediction remains a separate temporal validation task requiring trustworthy timestamps and a declared forecast horizon.

Index Terms—Internet of Vehicles, intrusion detection, intrusion prediction, Semantic Web, ontology, RDF, OWL, explainable AI, Random Forest, situation awareness.

I. INTRODUCTION

Connected and automated vehicles increasingly depend on communication among electronic control units, roadside infrastructure and external services. This evolution supports cooperative safety and intelligent mobility, but it also exposes vehicles to denial-of-service (DoS), spoofing, replay, fabrication, masquerade and other cyberattacks [1]–[5], [14]–[21]. Standards such as IEEE 1609.2, ISO/SAE 21434 and UNECE R155 address important communication and engineering controls, yet intrusion monitoring remains necessary because authenticated or syntactically valid behavior is not necessarily benign [9]–[12].

Existing IoV intrusion-detection systems (IDSs) range from timing/specification approaches to statistical, deep-learning, hybrid and federated architectures [22]–[31]. These methods can discriminate attack traffic, but model outputs may remain difficult to relate to affected assets, vulnerabilities and operational context. Explainability is particularly important in safety-sensitive systems because an analyst needs both a decision and the evidence that supports it.

Semantic Web technologies provide a complementary mechanism. RDF expresses linked facts, OWL formalizes classes and relations, SPARQL retrieves graph knowledge and rule languages can encode explicit inference conditions [32]–[43]. Security ontologies have long been used to represent attacks and cyber knowledge [44]–[50]. Xu et al. [49] demonstrated semantic ontology and user-defined rules for IoT network-security situation awareness, while recent automotive work such as KG-ID confirms that knowledge-graph-based intrusion detection is now an established research direction.

Accordingly, the novelty claimed here is not the use of an ontology by itself. The contribution is an IoV-specific integration of (i) source-traceable semantic context, (ii) rule-level explanations, and (iii) machine-learning evidence, while maintaining a strict distinction between current-event detection and future-horizon

prediction. The paper reports the results that are actually available from the thesis prototype and identifies the additional temporal experiment needed before advance prediction can be claimed.

II. RELATED WORK

A. IoV and Automotive Intrusion Detection

Automotive security studies have demonstrated practical attack surfaces and the weaknesses of the Controller Area Network (CAN) [14]–[21]. Timing-based IDSs exploit message periodicity [28], while deep neural and spatial-temporal models learn traffic representations [27], [29]. MTH-IDS combines multiple tiers of detection [22], and AI-oriented IoV architectures emphasize adaptive data-driven analysis [23], [24]. Federated approaches seek to train across distributed participants without centralizing all raw data [25], [30].

Dataset design strongly affects the validity of reported performance. CICIoV2024 provides labelled CAN evidence for DoS and spoofing scenarios [26]. The HCRL Car-Hacking dataset contains timestamped real-vehicle CAN traffic with DoS, fuzzy, gear and RPM spoofing attacks, while ROAD and VeReMi/VeReMi Extension provide complementary automotive and V2X evaluation settings. These datasets are useful future validation targets; however, they must not be conflated with the source-reported 1,200-row dataset used for the numerical experiment in this paper.

B. Semantic Security and Situation Awareness

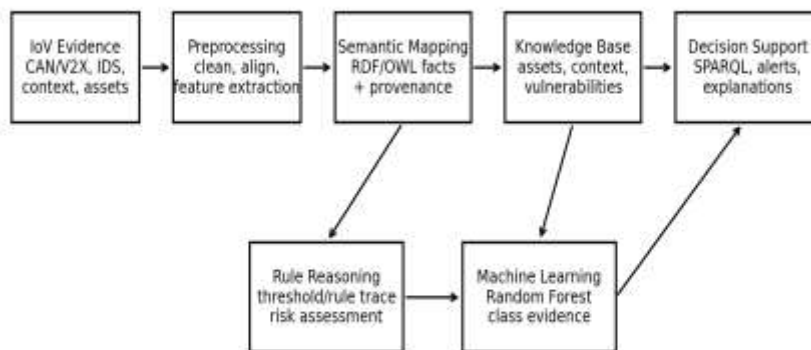
Situation awareness separates perception of relevant observations, comprehension of their meaning and projection of likely future states. This distinction is useful for intrusion research because a classifier that labels an already observed malicious record performs detection, whereas prediction requires evidence available before attack onset. Semantic representation supports the comprehension layer by connecting otherwise heterogeneous observations to assets and security knowledge.

Ontology-based security research includes attack ontologies, unified cybersecurity vocabularies and knowledge-graph construction [44]–[50]. The IoT model of Xu et al. [49] is especially relevant because it combines context, attack, vulnerability and network-flow concepts with user-defined rules. The proposed IoV framework adapts this principle to vehicle/ECU/RSU entities and combines semantic traces with a statistical classifier.

C. Explainability Gap

Post-hoc explainability methods such as SHAP can attribute a model decision to numerical features, but a semantic explanation answers a different question: how is an observation related to a vehicle asset, context, attack hypothesis or vulnerability? The framework therefore treats feature importance and semantic rule traces as complementary forms of explanation rather than interchangeable techniques.

III. PROPOSED FRAMEWORK



Present-event classification is evaluated here; future-horizon prediction requires time-ordered onset data.

Fig. 1. Proposed semantic-machine-learning IoV intrusion assessment architecture.

Figure 1 organizes the model into acquisition, preprocessing, semantic mapping, knowledge-base, reasoning/ML and decision-support responsibilities. These are logical layers and can execute in one Python process during controlled experiments. Data acquisition retains source and timestamp metadata where available. Preprocessing derives permitted features without using the target label. Semantic mapping instantiates RDF/OWL facts and provenance. The knowledge base connects evidence with documented asset, context, vulnerability and attack knowledge. Rule reasoning and machine learning then produce assessment evidence that can be retrieved through SPARQL or presented as an analyst-facing explanation.

A. Ontology and Knowledge Representation

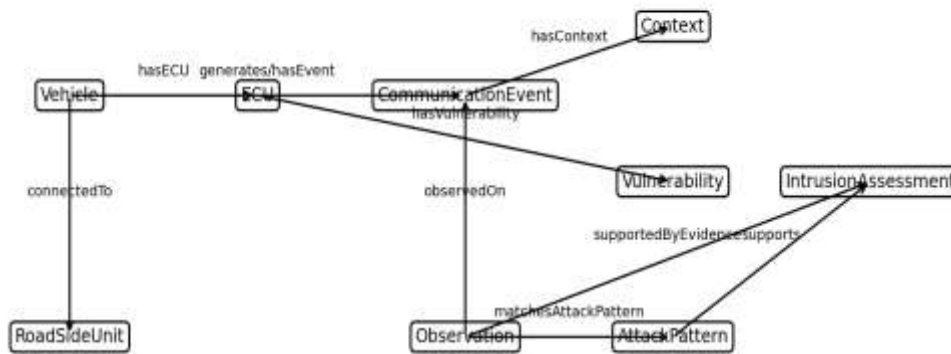


Fig. 2. Core ontology relationships used for traceable IoV intrusion assessment.

The stronger conceptual ontology contains Vehicle, ECU, RoadSideUnit, CommunicationEvent, Observation, Message, AttackPattern, Vulnerability, Context and IntrusionAssessment. Object properties include hasECU, observedOn, targetsECU, hasContext, hasVulnerability, matchesAttackPattern, supportedByEvidence and hasPredictedState. This design avoids asserting that an entire vehicle is permanently compromised when only one time-bounded event is suspicious. It also permits an assessment to point back to the observations that caused it.

The implemented prototype demonstrates a smaller subset: Vehicle, RoadSideUnit, CommunicationEvent, Intrusion, NormalCommunication, SuspiciousCommunication and AttackType, with hasCommunicationEvent and connectedTo object properties. Data properties store packet rate, failed-authentication count, beacon deviation and reputation score. The ontology was implemented in Python with Owlready2 [39].

B. Rule-Based Explainability

TABLE I. Demonstration rule catalogue used in the prototype.

Rule	Condition	Points	Explanation
R1	packet_rate > 70	2	High packet transmission rate
R2	failed_auth_count > 3	2	Repeated authentication failure
R3	beacon_deviation > 12	1	Abnormal beacon interval deviation
R4	reputation_score < 0.50	1	Low source reputation score

Decision	risk_score ≥ 3	—	Suspicious / Intrusion
----------	---------------------	---	------------------------

The rule layer is deliberately transparent. Each breached condition contributes a score and a plain-language reason. In the available implementation these conditions execute as Python logic; therefore, the results are described as semantic/rule-trace reasoning rather than as evidence of an executed SWRL reasoner. The thresholds are demonstration settings and require protocol-specific calibration before operational deployment.

C. Machine-Learning Component

Seven variables are used by the reported binary classifier: packet rate, average payload size, connection duration, speed variation, failed-authentication count, beacon-interval deviation and source-reputation score. Random Forest was selected because it can model nonlinear interactions, provides robust ensemble classification and exposes global feature importance [67]. For trees $T_1 \dots T_B$ and feature vector x , the predicted class is the majority vote: $\hat{y} = \text{mode}\{T_1(x), \dots, T_B(x)\}$.

The source script uses a stratified 75/25 train/test split with random_state=42. Thus, 900 of the 1,200 rows are used for training and 300 for testing. This is a row-level random split; it does not establish transfer to unseen vehicles, trips or captures.

IV. EXPERIMENTAL DESIGN

A. Evidence Status and Dataset

The implementation reports a dataset shape of $1,200 \times 8$ with seven predictors and the binary target intrusion_label. No missing values were printed. However, the available thesis evidence does not document the dataset publisher, capture dates, number of vehicles, trips, sampling procedure, or the derivation of trust/authentication variables. Consequently, the experiment is reported as a source-reported seven-feature dataset and is not labelled as CICIoV2024, HCRL Car-Hacking, ROAD or VeReMi.

TABLE II. Evidence register for the reported prototype.

Component	Reported evidence	Interpretation
Dataset	1,200 rows; 7 features + label	Source-reported; provenance unavailable
Split	900 train / 300 test	Stratified row-level random split
Ontology	Classes/properties + sample instances	Representation feasibility
Rules	4 thresholds + additive score	Worked explainability demonstration
ML	Random Forest + confusion matrix	Current-event binary classification

B. Evaluation Metrics

Taking intrusion as the positive class, $\text{accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN})$, $\text{precision} = \text{TP} / (\text{TP} + \text{FP})$, $\text{recall} = \text{TP} / (\text{TP} + \text{FN})$, and $\text{F1} = 2 \cdot \text{precision} \cdot \text{recall} / (\text{precision} + \text{recall})$. Because missed attacks are important in safety-sensitive systems, recall and the false-negative count are interpreted alongside accuracy.

V. RESULTS

A. Classification Performance

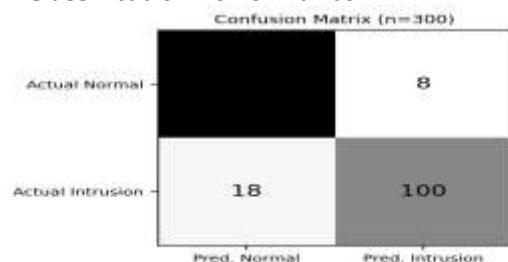
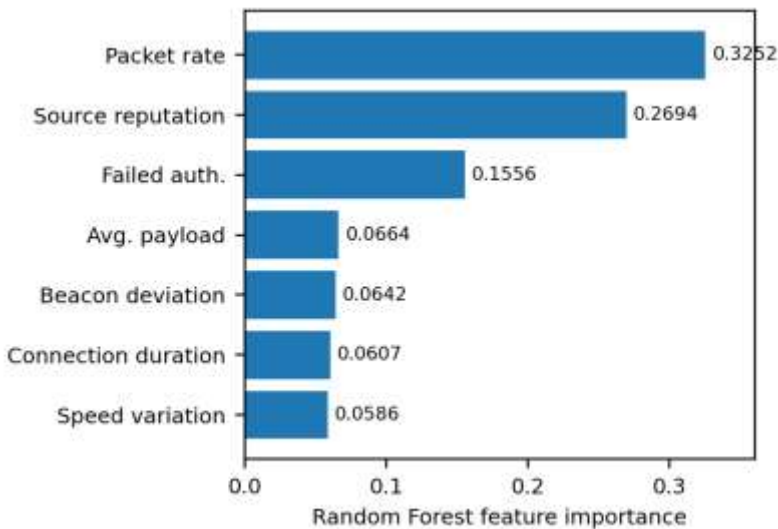


Fig. 3. Reported Random Forest confusion matrix.**TABLE III. Source-reported Random Forest performance.**

Metric	Value
Test records	300 (182 normal; 118 intrusion)
True negative / False positive	174 / 8
False negative / True positive	18 / 100
Accuracy	91.33%
Intrusion precision	92.59%
Intrusion recall	84.75%
Intrusion F1-score	88.50%

The classifier correctly labels 274 of 300 test records. Precision of 92.59% indicates that most flagged records are intrusion-labelled observations, while recall of 84.75% reveals a more important limitation: 18 of 118 intrusion-labelled records are assigned to the normal class. The reported figures are internally consistent after rounding, but they characterize only this split and should not be interpreted as verified deployment performance.

B. Feature Importance

**Fig. 4. Random Forest feature-importance values.**

Packet rate is the dominant feature (0.3252), followed by source reputation (0.2694) and failed-authentication count (0.1556). Together these three account for approximately 75% of the reported importance. The remaining values are average payload size 0.0664, beacon deviation 0.0642, connection duration 0.0607 and speed variation 0.0586. Global feature importance indicates what influenced the fitted ensemble overall; it is not a causal explanation of a specific event.

C. Semantic Explanation Example

TABLE IV. Rule trace for the worked communication event.

Input	Value	Rule result	Points
Packet rate	85.0	$85.0 > 70$	2
Failed authentication	4	$4 > 3$	2
Beacon deviation	14.2	$14.2 > 12$	1
Source reputation	0.41	$0.41 < 0.50$	1
Total	—	Suspicious / Intrusion	6

All four conditions fire for the worked event, producing risk score 6. The explanation is therefore inspectable: the analyst can see the exact observation, threshold and contribution associated with the assessment. This demonstrates traceability, not independently validated diagnostic reliability, because the example values and thresholds are illustrative rather than a documented real-vehicle attack trace.

VI. DISCUSSION

The main result is the coexistence of two evidence types. Random Forest provides statistical classification and global feature importance; the semantic layer provides explicit relationships and a human-readable rule trace. This is useful because an ML score alone does not identify the affected asset or explain how context and security knowledge relate to the observation.

The result should also be bounded carefully. The present experiment does not show that adding the ontology improves Random Forest accuracy, because no ML-only versus semantic+ML ablation has yet been executed. Likewise, the ontology example demonstrates representation but the supplied results do not report reasoner consistency checks, competency-question pass rates, inference counts or provenance-completeness measurements. Those tests are required before claiming full ontology validation.

Recent work further sharpens the contribution. Federated IoV IDS research addresses distributed privacy-aware learning [25], while KG-ID demonstrates knowledge-graph-based automotive intrusion detection. Explainable deep-learning frameworks use feature-attribution techniques such as SHAP. The proposed framework is complementary: it emphasizes a traceable path from event evidence to IoV entities and explicit security rules, while retaining ML evidence as a separate layer.

A. Detection Versus Prediction

The paper title uses intrusion prediction because that is the thesis objective and target architecture; nevertheless, the reported numerical experiment is current-event classification. A defensible advance-prediction experiment must define an attack onset t_a , a decision time $t < t_a$ and forecast horizon H , then use only information available at t to predict whether an intrusion begins in $[t, t+H]$. Evaluation should report onset recall, false-alert rate and warning lead time. Packet-labelled datasets without trustworthy onset timing cannot establish this claim.

B. Threats to Validity

Four threats are material. First, dataset provenance is unavailable for the reported 1,200 rows. Second, row-level random splitting may overestimate generalization when nearby vehicular observations are correlated. Third, the four rule thresholds have no documented calibration or sensitivity analysis. Fourth, the semantic prototype has not yet been evaluated with formal consistency, competency-query and reasoning benchmarks. These limitations motivate the validation programme rather than invalidate the architectural contribution.

VII. FUTURE VALIDATION PROGRAMME

A stronger final experiment should use named, documented benchmarks. CICIoV2024 is suitable for CAN attack classification [26]; HCRL Car-Hacking offers timestamped real-vehicle CAN traffic; ROAD supports cross-capture automotive evaluation; and VeReMi Extension supports V2X misbehavior scenarios. Where attack-onset metadata is unavailable, the corresponding experiment should be reported as detection rather than prediction.

Three ablations are particularly important: (1) ML-only, (2) rule/semantic-only, and (3) fused semantic+ML. The evaluation should also hold out complete vehicles, trips or captures where the dataset permits, calibrate thresholds only on development data, test normal/malicious/boundary rule cases, execute ontology consistency and competency questions, and measure mapping/reasoning latency. SHAP can additionally be incorporated to compare statistical feature attribution with semantic rule explanations.

VIII. CONCLUSION

This paper presented an explainable Semantic Web and machine-learning framework for IoV intrusion assessment. The framework models IoV assets, communication events, observations, context, vulnerabilities and attack knowledge as linked semantic entities, while rule traces expose why a selected event is considered suspicious. Random Forest supplies a complementary data-driven classification layer. On the source-reported 300-record test set, the classifier achieved 91.33% accuracy, 92.59% intrusion precision, 84.75% recall and 88.50% F1-score. Packet rate, source reputation and failed-authentication count were the three most influential variables.

The results support the feasibility of explainable current-event intrusion assessment, but they do not yet prove that semantic fusion increases classification accuracy or that attacks are predicted before onset. The next empirical step is therefore a reproducible evaluation on documented IoV datasets, with ontology validation, ablation, capture-aware holdouts and a separate time-horizon experiment for genuine advance prediction.

IX. DETAILED COMPARATIVE ANALYSIS

Table V positions the proposed framework against representative families of IoV and automotive IDS. The comparison is intentionally capability-oriented rather than a numerical ranking because published accuracy values are not directly comparable when datasets, attack definitions, partitions and preprocessing differ. A timing IDS, for example, may be evaluated on CAN periodicity, whereas a federated model may target distributed privacy and a semantic model may target knowledge integration. The relevant question is therefore which evidence each family consumes, which output it produces, and whether the output is traceable to domain knowledge.

TABLE V. Capability-oriented comparison with representative related approaches.

Approach	Primary evidence	Strength	Boundary relative to this work
Timing/specification IDS [28]	CAN inter-arrival behavior	Directly testable timing deviations	Limited semantic context and asset relationships
Deep neural IDS [27]	CAN feature vectors	Learns nonlinear attack patterns	Prediction rationale may be opaque
Spatial-temporal DL [29]	Sequential IVN traffic	Captures temporal/spatial structure	Requires separate explanation mechanism
MTH-IDS [22]	Multi-tier traffic evidence	Hybrid detection stages	Semantic provenance not its central objective
Federated IoV IDS [25], [30]	Distributed participant data	Privacy-aware collaborative learning	Context/ontology fusion remains separate
Semantic IoT NSSA [49]	Context, attack, vulnerability, flow	Explicit ontology + user rules	Not specifically evaluated as this IoV ML prototype
KG-ID [50]	In-vehicle knowledge graph	Automotive graph-based IDS	Confirms graph IDS prior art; novelty must be more specific
Proposed framework	IoV observations + semantic context + ML	Rule trace plus statistical evidence	Current experiment still needs ablation and temporal prediction

The table clarifies two points. First, semantic modelling is not presented as a replacement for high-performing statistical detection. Rather, it supplies an explicit vocabulary and evidence graph in which a classifier output can be related to the vehicle, component, communication event and available context. Second, the proposed contribution must be evaluated at more than one level. Classification metrics test the statistical component; competency questions and consistency checks test the ontology; rule cases test the explanation logic; and temporal experiments test any claim of advance prediction.

This separation is important for reproducibility. If a hybrid system reports only a single accuracy value, it is impossible to determine whether an ontology, a rule layer, a new feature transformation or the classifier itself produced the change. The planned ablation therefore treats the semantic knowledge base, deterministic rules and statistical classifier as separable components before evaluating their fusion.

X. ONTOLOGY ENGINEERING AND COMPETENCY QUESTIONS

Ontology engineering begins with a scoped conceptualization rather than with arbitrary class creation [33], [40]–[42]. The intended domain is intrusion-related IoV knowledge required to interpret a time-bounded communication event. Vehicle and ECU represent assets; RoadSideUnit represents roadside infrastructure; CommunicationEvent and Observation represent recorded activity; Context captures time-appropriate

operational conditions; Vulnerability and AttackPattern represent documented security knowledge; and IntrusionAssessment represents the system output. The design deliberately separates an observation from an assessment so that raw evidence is not silently converted into a security conclusion.

The identity policy is equally important. Repeated CAN identifiers do not automatically denote the same event or a unique transmitting ECU. A message identifier is therefore treated as an observed field unless a documented mapping supports a stronger assertion. Similarly, a vulnerability association is stored only when a verified asset-to-vulnerability mapping exists. Unknown facts remain unknown rather than being synthesized merely to enrich the graph. This provenance discipline is essential if semantic explanations are to be auditable.

Competency questions provide a practical specification for the ontology. Representative questions include: Which vehicle or ECU is associated with the suspicious event? Which observations support the current intrusion assessment? Which attack pattern is compatible with those observations? Which contextual facts were valid at the assessment time? Does the affected asset have a documented vulnerability relevant to the hypothesis? Which rule conditions fired, and which evidence values caused them to fire? These questions can be translated into SPARQL queries once the ontology and instance mappings are fixed.

TABLE VI. Example competency questions for semantic validation

Competency question	Required graph path	Expected validation evidence
CQ1: Which asset is associated with Event e?	Event → target/vehicle/ECU	Correct asset returned or unknown
CQ2: What evidence supports Assessment a?	Assessment → supportedByEvidence → Observation	All contributing observations returned
CQ3: Which attack pattern matches?	Observation → matchesAttackPattern → AttackPattern	Only rule-supported hypothesis returned
CQ4: What context was valid at t?	Event → hasContext → Context	Time-consistent context returned
CQ5: Why was the event suspicious?	Assessment → evidence + rule trace	Threshold/value/reason visible
CQ6: What is missing?	Required property/provenance checks	Missing facts remain explicitly absent

Formal validation should include reasoner consistency, satisfiability of core classes, intended domain/range behavior, duplicate-individual checks, and positive/negative competency-query cases. The current source implementation demonstrates class/property construction and instance creation, but those stronger ontology-quality results have not yet been executed and are therefore proposed rather than claimed.

XI. DATA PIPELINE AND FEATURE SEMANTICS

A semantic ML system must define not only feature names but also how each value is measured. Packet rate requires a timestamped interval and grouping key; average payload size requires a protocol and aggregation window; connection duration requires a defensible session boundary; speed variation requires an authenticated or otherwise reliable speed source; failed-authentication count requires an authentication mechanism and observation interval; beacon deviation requires an expected beacon schedule; and source reputation requires a documented trust model. Without these definitions, similarly named columns can represent materially different phenomena.

TABLE VII. Semantic interpretation of the seven reported classifier features.

Feature	Semantic interpretation	Reproducibility requirement
packet_rate	Message/packet frequency in a window	Timestamp, window length, grouping entity
avg_payload_size	Mean payload in aggregation window	Protocol, units, missing-byte policy

connection_duration	Observed session duration	Session start/end definition
speed_variation	Change/dispersion in vehicle speed	Speed source, time alignment, unit
failed_auth_count	Authentication failures in window	Authentication source and counting policy
beacon_interval_deviation	Deviation from expected periodicity	Expected interval and deviation formula
source_reputation_score	Trust/reputation estimate	Scale, update policy and provenance

The ontology can store these measurements as observations linked to their event and provenance. This enables the system to distinguish a numerical value from the interpretation attached to it. For example, packet_rate=85 has little meaning without the relevant interval and message family. A future implementation should therefore represent measurement metadata or link to a sensor/observation model such as SSN where appropriate [43].

Preprocessing must also respect the intended prediction time. For current-event classification, features may be computed from the observed event window. For advance prediction, no feature may include packets or labels occurring after the decision time. This restriction prevents target leakage and is particularly important for rolling rates, reputation updates and temporal anomaly scores.

XII. FUSION AND DECISION LOGIC

The proposed hybrid architecture does not require that semantic and ML evidence be collapsed into a single opaque number. A more auditable design stores the classifier probability or class as one observation, stores deterministic rule outcomes as separate observations, and lets an explicit policy create the final IntrusionAssessment. Such a policy can require agreement, use a calibrated weighted score, or escalate conflicting evidence for analyst review. Whichever policy is selected must be frozen before testing on the final holdout.

For the current prototype, the demonstrated rule score is additive: R1 and R2 contribute two points each, while R3 and R4 contribute one point each; score ≥ 3 produces a suspicious/intrusion assessment. This is useful pedagogically because the reasoning path is visible. Operational use would require calibration on development data and sensitivity analysis around each threshold. The analysis should report how false positives and false negatives change as thresholds vary, rather than selecting values after inspecting the final test set.

Conflict handling is also necessary. A future event may receive a high ML intrusion probability while no semantic rule fires because contextual facts are missing. Conversely, a high-risk semantic rule may fire while the statistical classifier returns normal. Rather than hiding such disagreement, the knowledge graph can represent the two evidence sources and the final policy outcome. This makes uncertainty and missingness part of the explanation.

XIII. VALIDATION PROTOCOL FOR THE FINAL THESIS EXPERIMENT

A publication-quality final evaluation should lock the data manifest, preprocessing scripts, random seeds and partitions before model comparison. When possible, entire captures, trips or vehicles should be held out to reduce dependence between training and test observations. Hyperparameters and rule thresholds must be selected using training/development data only. The untouched test set is then used once for the principal result, with confidence intervals or repeated group-aware splits where the dataset supports them.

The semantic artifact should be evaluated separately from the classifier. A minimal semantic test suite contains ontology consistency, competency questions, positive and negative rule cases, missing-field behavior, provenance checks and execution time. The ML test suite contains confusion matrices, per-class precision/recall/F1, false-positive and false-negative rates, and precision-recall analysis when class imbalance is substantial. A fusion test then compares the combined system with its reduced variants.

TABLE VIII. Recommended controlled experiments for validation.

Experiment	Configuration	Primary outcome
E1	Random Forest only	Classification baseline

E2	Rule/semantic assessment only	Rule precision/recall + explanation coverage
E3	RF + semantic/rule fusion	Incremental value over E1/E2
E4	E3 without context links	Ablation of contextual semantics
E5	E3 without vulnerability/attack knowledge	Ablation of domain knowledge
E6	Capture/vehicle-held-out evaluation	Generalization beyond random rows
E7	Temporal pre-onset evaluation	Lead time, onset recall, false-alert rate

This protocol prevents a common interpretive error: assuming that a semantically richer system is automatically a more accurate classifier. The semantic layer may add value through explanation, interoperability and contextual discrimination even if raw classification accuracy changes little. Conversely, if fusion improves F1 or reduces false alarms, an ablation is required to establish which component caused the improvement.

XIV. TEMPORAL INTRUSION-PREDICTION FORMULATION

To convert the architecture from current-event detection to empirically demonstrated prediction, the target variable must be future-facing. Let t_a denote the onset time of a verified intrusion episode and t the decision time. For horizon $H > 0$, define $y(t, H) = 1$ when a new intrusion begins in $[t, t + H]$ and $y(t, H) = 0$ otherwise. Every feature used at t must be computable using data with timestamps $\leq t$. Windows that cross t_a or use the future attack label are prohibited.

The temporal evaluation should report more than ordinary packet-level accuracy. Onset recall measures the fraction of attack episodes that receive at least one valid warning before onset. Lead time $t_a - t_w$ measures how early the first correct warning t_w occurs. False-alert rate should be normalized by time or decision opportunities in non-attack periods. Precision remains useful because frequent premature alarms can make a warning system operationally unusable even when onset recall is high.

Different horizons can reveal the practical boundary of predictability. A short horizon may provide strong discrimination but little reaction time; a long horizon may provide more warning but lower precision. Results should therefore be presented as a horizon-performance curve rather than a single unqualified prediction accuracy. Datasets that contain only independent attack-labelled packets without trustworthy episode boundaries are unsuitable for this experiment and should remain in the detection track.

XV. REPRODUCIBILITY AND IMPLEMENTATION CONSIDERATIONS

The current prototype uses Python data-processing libraries, Owlready2 for ontology objects, RDFLib-style graph construction and scikit-learn for Random Forest classification. A reproducible release should additionally record exact package versions, operating-system information, hardware, random seeds, ontology version, rule catalogue version and checksums of every raw dataset. Derived files should be regenerated by scripts rather than manually edited.

For public datasets, a manifest should record publisher, download date, license, checksum, raw-file inventory and label definitions. CICIoV2024, HCRL Car-Hacking, ROAD and VeReMi Extension differ substantially in capture environment and available fields; therefore, a common feature should be computed only when its definition can be preserved. Missing source reputation or authentication evidence should remain unavailable instead of being imputed as if it were measured.

Runtime is another important outcome. IoV decisions may be time-sensitive, and semantic reasoning introduces mapping and query overhead in addition to classifier inference. The final evaluation should therefore measure preprocessing latency, graph-instantiation time, reasoning/query time, classifier inference time and end-to-end assessment latency. Graph size and memory use should be reported for increasing numbers of events to assess scalability.

XVI. IMPLICATIONS FOR EXPLAINABLE IoV SECURITY

The framework supports a layered notion of explanation. At the data level, the analyst sees measured values such as packet rate or authentication failures. At the model level, feature importance or a future SHAP

analysis can show how the statistical classifier uses numerical variables. At the knowledge level, ontology relations and rule traces explain how the event is connected to assets, context, vulnerabilities and attack hypotheses. These levels answer different questions and should be reported separately.

This layered design can also improve communication between engineering roles. A data scientist may inspect classifier evidence, a vehicle-security engineer may inspect ECU and attack relationships, and an analyst may inspect the final rule trace and provenance. The knowledge graph becomes a shared representation rather than a claim that symbolic reasoning alone can solve the intrusion-detection problem.

The approach remains bounded by evidence quality. A detailed explanation constructed from incorrect context is still incorrect. Therefore, provenance, uncertainty and missing-data policy are not secondary metadata; they are part of trustworthy explainability. The final system should expose when a conclusion depends on simulated context, a heuristic reputation value or an unverified asset mapping.

REFERENCES

- [1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [2] M. Gerla, E.-K. Lee, G. Pau, and U. Lee, "Internet of Vehicles: From intelligent grid to autonomous cars and vehicular clouds," in *Proc. IEEE WF-IoT*, 2014, pp. 241–246.
- [3] H. Hartenstein and K. P. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Commun. Mag.*, vol. 46, no. 6, pp. 164–171, 2008.
- [4] J. Contreras-Castillo, S. Zeadally, and J. A. Guerrero-Ibañez, "Internet of Vehicles: Architecture, protocols, and security," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3701–3709, 2018.
- [5] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, "A comprehensive survey on vehicular ad hoc network," *J. Netw. Comput. Appl.*, vol. 37, pp. 380–392, 2014.
- [6] IEEE Std 802.11p-2010, *Wireless Access in Vehicular Environments*, 2010.
- [7] ETSI EN 302 663 V1.3.1, "ITS; Access layer specification for ITS operating in the 5 GHz frequency band," 2020.
- [8] 3GPP TS 23.287, "Architecture enhancements for 5G System to support V2X services," 2023.
- [9] IEEE Std 1609.2-2022, *Security Services for Applications and Management Messages*, 2022.
- [10] ETSI TS 103 097 V2.1.1, "ITS; Security; Security header and certificate formats," 2021.
- [11] ISO/SAE 21434:2021, *Road Vehicles—Cybersecurity Engineering*, 2021.
- [12] UNECE, "UN Regulation No. 155: Cyber security and cyber security management system," 2021.
- [13] SAE International, *Cybersecurity Guidebook for Cyber-Physical Vehicle Systems*, SAE J3061, 2021.
- [14] K. Koscher et al., "Experimental security analysis of a modern automobile," in *Proc. IEEE Symp. Security and Privacy*, 2010, pp. 447–462.
- [15] S. Checkoway et al., "Comprehensive experimental analyses of automotive attack surfaces," in *Proc. USENIX Security Symp.*, 2011, pp. 77–92.
- [16] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," *Black Hat USA*, 2015.
- [17] T. Hoppe, S. Kiltz, and J. Dittmann, "Security threats to automotive CAN networks: Practical examples and selected short-term countermeasures," in *Proc. SAFECOMP*, 2008, pp. 235–248.
- [18] H. Taslimasa et al., "Security issues in Internet of Vehicles (IoV): A comprehensive survey," *Internet of Things*, vol. 24, Art. no. 100809, 2023.
- [19] F. Sakiz and S. Sen, "A survey of attacks and detection mechanisms on intelligent transportation systems: VANETs and IoV," *Ad Hoc Networks*, vol. 61, pp. 33–50, 2017.
- [20] G. Loukas et al., "A taxonomy and survey of cyber-physical intrusion detection approaches for vehicles," *Ad Hoc Networks*, vol. 84, pp. 124–147, 2019.
- [21] E. Aliwa, O. Rana, C. Perera, and P. Burnap, "Cyberattacks and countermeasures for in-vehicle networks," *ACM Comput. Surv.*, vol. 54, no. 1, pp. 1–37, 2021.
- [22] L. Yang, A. Moubayed, and A. Shami, "MTH-IDS: A multitiered hybrid intrusion detection system for Internet of Vehicles," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 616–632, 2022.
- [23] T. Alladi, V. Kohli, V. Chamola, F. R. Yu, and M. Guizani, "AI-empowered intrusion detection architecture for the Internet of Vehicles," *IEEE Wireless Commun.*, vol. 28, no. 3, pp. 144–149, 2021.
- [24] L. Yang and A. Shami, "A transfer learning and optimized CNN based intrusion detection system for Internet of Vehicles," in *Proc. IEEE ICC*, 2022, pp. 2774–2779.

- [25] K. Huang, R. Xian, M. Xian, H. Wang, and L. Ni, "A comprehensive intrusion detection method for the Internet of Vehicles based on federated learning architecture," *Computers & Security*, vol. 147, Art. no. 104067, 2024.
- [26] E. C. P. Neto et al., "CICIoV2024: Advancing realistic IDS approaches against DoS and spoofing attack in IoV CAN bus," *Internet of Things*, vol. 26, Art. no. 101209, 2024.
- [27] M.-J. Kang and J.-W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security," *PLOS ONE*, vol. 11, no. 6, Art. no. e0155781, 2016.
- [28] H. M. Song, H. R. Kim, and H. K. Kim, "Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network," in *Proc. ICOIN*, 2016, pp. 63–68.
- [29] W. Lo et al., "A hybrid deep learning based intrusion detection system using spatial-temporal representation of in-vehicle network traffic," *Vehicular Communications*, vol. 35, Art. no. 100471, 2022.
- [30] J. Alsamiri and K. Alsubhi, "Federated learning for intrusion detection systems in Internet of Vehicles: A general taxonomy, applications, and future directions," *Future Internet*, vol. 15, no. 12, Art. no. 403, 2023.
- [31] B. Lampe and W. Meng, "can-train-and-test: A curated CAN dataset for automotive intrusion detection," *Computers & Security*, 2024.
- [32] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," *Scientific American*, vol. 284, no. 5, pp. 34–43, 2001.
- [33] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge Acquisition*, vol. 5, no. 2, pp. 199–220, 1993.
- [34] R. Studer, V. R. Benjamins, and D. Fensel, "Knowledge engineering: Principles and methods," *Data & Knowledge Engineering*, vol. 25, nos. 1–2, pp. 161–197, 1998.
- [35] W3C, "RDF 1.1 Concepts and Abstract Syntax," W3C Recommendation, 2014.
- [36] W3C, "OWL 2 Web Ontology Language Document Overview," 2nd ed., W3C Recommendation, 2012.
- [37] W3C, "SPARQL 1.1 Query Language," W3C Recommendation, 2013.
- [38] I. Horrocks et al., "SWRL: A Semantic Web Rule Language combining OWL and RuleML," W3C Member Submission, 2004.
- [39] J.-B. Lamy, "Owlready: Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies," *Artif. Intell. Med.*, vol. 80, pp. 11–28, 2017.
- [40] N. F. Noy and D. L. McGuinness, "Ontology development 101: A guide to creating your first ontology," Stanford KSL, Tech. Rep. KSL-01-05, 2001.
- [41] M. Uschold and M. Grüninger, "Ontologies: Principles, methods and applications," *Knowledge Engineering Review*, vol. 11, no. 2, pp. 93–136, 1996.
- [42] M. Fernández-López, A. Gómez-Pérez, and N. Juristo, "METHONTOLOGY: From ontological art towards ontological engineering," in *Proc. AAAI Spring Symp. Ontological Engineering*, 1997, pp. 33–40.
- [43] A. Haller et al., "Semantic Sensor Network Ontology," W3C Recommendation, 2017.
- [44] J. Undercoffer, A. Joshi, and J. Pinkston, "Modeling computer attacks: An ontology for intrusion detection," in *Proc. RAID*, 2003, pp. 113–135.
- [45] J. Wang, M. M. Guo, and J. Camargo, "An ontological approach to computer system security," *Information Security Journal*, vol. 19, no. 2, pp. 61–73, 2010.
- [46] M. Iannacone et al., "Developing an ontology for cyber security knowledge graphs," in *Proc. CISRC*, 2015, pp. 1–4.
- [47] Z. Syed, A. Padia, T. Finin, L. Mathews, and A. Joshi, "UCO: A unified cybersecurity ontology," in *Proc. AAAI Workshop on AI for Cyber Security*, 2016.
- [48] L. Obrst, P. Chase, and R. Markeloff, "Developing an ontology of the cyber security domain," in *Proc. STIDS*, 2012, pp. 49–56.
- [49] G. Xu, Y. Cao, Y. Ren, X. Li, and Z. Feng, "Network security situation awareness based on semantic ontology and user-defined rules for Internet of Things," *IEEE Access*, vol. 5, pp. 21046–21056, 2017.
- [50] H. Sun, J. Wang, J. Weng, and W. Tan, "KG-ID: Knowledge graph-based intrusion detection on in-vehicle network," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 4, pp. 4988–5000, 2025.